Flash code description

FLASH
code description

by Thomas Kocher, 08.03.2006
thomas.kocher@alumni.ethz.ch

This document describes the basic principles of continuum mechanics and the finite element method (FEM) as far as they are necessary to understand their implementation in the FLASH code. The text presented in the following forms part of the ETH thesis no. 16413,

'Flanking structure and single layer fold development in isotropic and anisotropic rock',

that was submitted in Dec. 2005 by the author.

The following pages are intended to support readers of publications that contain results obtained from the FLASH code in understanding and verifying the results.
The text contains the basics of the FE method as well as print-outs of Maple code describing the anisotropy formulation. Please be aware that for the Software, and analogously for the theory described in this text, the below disclaimer and limitations apply. Please direct any questions or comments to the above
email-address.

A.2 A short outline of the Finite Element Method

In this section, the fundamentals of the finite element method are presented as far they are necessary to understand the implementation in the FLASH code that was used in this thesis to investigate different geomechanical problems. A vast amount of literature exists on the techniques of finite elements (e.g Cuvelier et al., 1986; Buchanan, 1994; Kwon and Bang, 2000; Zienkiewicz and Taylor, 2000; Smith and Griffith, 2004).

A concise summary of the basic equations is given here to enable a clear understanding of the code that was used, which is crucial for the interpretation of the numerical results. Symbolic and component notation are mixed in this section. The notation convention follows the definition in subsection A.1.2.

A.2.1 Governing physical equations

The finite element code FLASH solves the equations for mass and momentum balance, Eqs. (A.4) and (A.7):

$$\frac{D\rho}{Dt} + \rho \boldsymbol{\nabla}_{\mathbf{x}} \cdot \mathbf{v} = 0 \tag{A.30}$$

$$\rho \frac{D\mathbf{v}}{Dt} = \boldsymbol{\nabla}_{\mathbf{x}} \cdot \mathbf{T} + \rho \mathbf{k}. \tag{A.31}$$

Assuming that the material is incompressible, Eq. (A.30) can be simplified to

$$\boldsymbol{\nabla}_{\mathbf{x}} \cdot \mathbf{v} = 0. \tag{A.32}$$

For geological applications on a large time scale, the velocities involved are usually very low. For that reason, inertial forces play a minor role and can be neglected by setting the local derivative on the left hand side in Eq. (A.31) equal to zero:

$$\boldsymbol{\nabla}_{\mathbf{x}} \cdot \mathbf{T} + \rho \mathbf{k} = 0. \tag{A.33}$$

Written out for the two-dimensional case, including gravity $g = -9.81$ m/s^2 as a body force, and denoting the components of the stress tensor $\mathbf{T}$ as σ_{ij}, the equations for the

mass and linear momentum balance read:

$$(A.32) \rightarrow \quad \frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} = 0 \tag{A.34}$$

$$(A.33) \rightarrow \quad \frac{\partial \sigma_{xx}}{\partial x} + \frac{\partial \sigma_{xy}}{\partial y} = 0 \tag{A.35}$$

$$(A.33) \rightarrow \quad \frac{\partial \sigma_{xy}}{\partial x} + \frac{\partial \sigma_{yy}}{\partial y} + \rho g = 0 \tag{A.36}$$

These three equations have to be completed with rheological equations in order to equal the number of equations to the number of unknowns, and to specify the rheological behaviour of the material. For the case of an incompressible power-law fluid, the complete system of equations then reads:

$$\frac{\partial \sigma_{xx}}{\partial x} + \frac{\partial \sigma_{xy}}{\partial y} = 0 \tag{A.37}$$

$$\frac{\partial \sigma_{xy}}{\partial x} + \frac{\partial \sigma_{yy}}{\partial y} + \rho g = 0 \tag{A.38}$$

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} = 0 \tag{A.39}$$

$$\sigma_{xx} = -p + 2\mu_{eff}\dot{\epsilon}_{xx} \tag{A.40}$$

$$\sigma_{yy} = -p + 2\mu_{eff}\dot{\epsilon}_{yy} \tag{A.41}$$

$$\sigma_{xy} = \mu_{eff}\dot{\epsilon}_{xy}, \tag{A.42}$$

where μ_{eff} is again the effective viscosity (Eq. (A.28)). Note that the stresses σ_{ij} are given in complete and not in deviatoric form. Eqs. (A.37) to (A.42) can be rewritten in matrix form:

$$\begin{bmatrix} \frac{\partial}{\partial x} & 0 & \frac{\partial}{\partial y} \\ 0 & \frac{\partial}{\partial y} & \frac{\partial}{\partial x} \end{bmatrix} \begin{pmatrix} \sigma_{xx} \\ \sigma_{yy} \\ \sigma_{xy} \end{pmatrix} = \begin{pmatrix} 0 \\ -\rho g \end{pmatrix}, \tag{A.43}$$

$$\begin{pmatrix} \sigma_{xx} \\ \sigma_{yy} \\ \sigma_{xy} \end{pmatrix} = \begin{pmatrix} -p \\ -p \\ 0 \end{pmatrix} + \begin{bmatrix} 2\mu & 0 & 0 \\ 0 & 2\mu & 0 \\ 0 & 0 & \mu \end{bmatrix} \begin{pmatrix} \dot{\epsilon}_{xx} \\ \dot{\epsilon}_{yy} \\ \dot{\epsilon}_{xy} \end{pmatrix}, \tag{A.44}$$

where strain rates $\dot{\epsilon}_{ij}$ are related to velocitys:

$$\begin{pmatrix} \dot{\epsilon}_{xx} \\ \dot{\epsilon}_{yy} \\ \dot{\epsilon}_{xy} \end{pmatrix} = \begin{pmatrix} \frac{\partial u}{\partial x} \\ \frac{\partial v}{\partial y} \\ \frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \end{pmatrix}. \tag{A.45}$$

Combining Eq. (A.43) to (A.45), the full system can now be rewritten in symbolic notation:

$$-\mathbf{Ap} + \mathbf{AMA^T d} = \mathbf{f} \tag{A.46}$$

$$\mathbf{Aed} = 0, \tag{A.47}$$

where

$$\mathbf{A} = \begin{bmatrix} \frac{\partial}{\partial x} & 0 & \frac{\partial}{\partial y} \\ 0 & \frac{\partial}{\partial y} & \frac{\partial}{\partial x} \end{bmatrix} \qquad \mathbf{M} = \begin{bmatrix} 2\mu & 0 & 0 \\ 0 & 2\mu & 0 \\ 0 & 0 & \mu \end{bmatrix} \tag{A.48}$$

$$\mathbf{d} = \begin{pmatrix} u \\ v \end{pmatrix} \qquad \mathbf{p} = \begin{pmatrix} p \\ p \\ 0 \end{pmatrix} \qquad \mathbf{f} = \begin{pmatrix} 0 \\ -\rho g \end{pmatrix} \qquad \mathbf{e} = \begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix}. \tag{A.49}$$

A.2.2 Discretisation of the unknown parameters

The unknowns velocities u, v and the pressures p are discretized on a finite element grid using isoparametric shape functions (see Kwon and Bang, 2000, for a thorough introduction):

$$u = H_i(\zeta, \eta)u_i \qquad v = H_i(\zeta, \eta)v_i \qquad p = N_i(\zeta, \eta)p_i, \tag{A.50}$$

where H_i and N_i are two different sets of shape functions (and ζ, η the isoparametric coordinates), and u_i, v_i and p_i the velocities and pressure coefficients at the respective nodes of the element. The elements used in FLASH are either 7-node triangular Crouzeix-Raviart elements, with an enriched quadratic discretisation for velocities, and a linear 3-degrees of freedom pressure approximation ($P_2^+ - P_1$), or 9-node bilinear quadratic elements with the same linear pressure approximation as for the triangular elements (Cuvelier et al., 1986). These elements proved to give very accurate results for both velocity/stress and pressure fields for the given problem setups, as is demonstrated in section A.5.

A.2.3 The method of weighted residuals

The discretisation of the unknown variables u, v and p causes the differential equations for linear momentum and mass balance to be no longer exactly fulfilled; a residual remains that depends on the accuracy of the approximation. By multiplying the partial differential equations (PDE) with an arbitrary weighting or test function w, integrating over the element domain Ω and requiring the resulting functional to be equal to 0, the

numerical solution is forced to fulfil the PDE with a residual that is 'zero on average' over the domain. The weighted residual form of the linear momentum balance equation (A.46) then reads:

$$-\int_{\Omega_{el}} w(\mathbf{Ap})d\Omega + \int_{\Omega_{el}} w(\mathbf{AMA^T d})d\Omega - \int_{\Omega_{el}} w\mathbf{f}d\Omega = 0. \tag{A.51}$$

(If flux boundary conditions were present, they would have to be included in the weighted residual formulation as well (Zienkiewicz and Taylor, 2000, vol. 1)).

The second term in Eq. (A.51) contains a second order derivative of the velocities $\mathbf{d}$. This means that the second order derivative of the velocities, and hence of the shape functions H_i, must be of a finite, nonzero value, so that the integral can be evaluated at all points within the domain. A method to relax this condition for the velocity shape functions is to formulate the weighted residual in a weak form. Through this procedure, the requirements (with respect to continuity) on the shape functions H_i are relaxed and 'shifted' to the weighting function w.

Weak formulation of the weighted residual

The weighted residual can be reformulated by applying the Green's Theorem – the generalization of 1-D partial integration to higher dimensions – to the first two terms in Eq. (A.51):

$$-\int_{\Omega_{el}} w(\mathbf{Ap})d\Omega = \int_{\Omega_{el}} (w\mathbf{A})\mathbf{p} - \oint_{\Gamma} \ldots \tag{A.52}$$

$$\int_{\Omega_{el}} w(\mathbf{AMA^T d})d\Omega = -\int_{\Omega_{el}} (w\mathbf{A})\mathbf{MA^T d} + \oint_{\Gamma} \ldots . \tag{A.53}$$

The third term describing the body force is left as is. If no Neumann (or flux) boundary conditions are applied, and if the Dirichlet boundary conditions are enforced (by imposing them on the stiffness matrix), the boundary integrals can be dropped (see Zienkiewicz and Taylor, 2000, vol. 1, pg. 45). The force balance can then be written in the weak formulation:

$$\int_{\Omega_{el}} (w\mathbf{A})\mathbf{p} - \int_{\Omega_{el}} (w\mathbf{A})\mathbf{MA^T d} - \int_{\Omega_{el}} w\mathbf{f}d\Omega = 0 \tag{A.54}$$

A.2.4 Galerkin's formulation of the weighted residual

The weighting function w is only restricted so far by the requirement that $w = 0$ on the boundaries of the domain (then the boundary integrals can be dropped in Eq. (A.52) and (A.53)). Following Galerkin's approach (Kwon and Bang, 2000) and replacing the weighting function w by the shape functions of the velocities, h_i ($\mathbf{h} = (H_1, H_2, \ldots)$), a

system of equations can be set up for each element:

$$-\int_{\Omega_{el}} \mathbf{B^T N} d\Omega \mathbf{p} + \int_{\Omega_{el}} \mathbf{B^T M B} d\Omega \mathbf{d} + \int_{\Omega_{el}} \mathbf{H f} d\Omega = 0, \tag{A.55}$$

where

$$\mathbf{B} = \begin{bmatrix} \frac{\partial H_1}{\partial x} & 0 & \frac{\partial H_2}{\partial x} & 0 & \dots \\ 0 & \frac{\partial H_1}{\partial y} & 0 & \frac{\partial H_2}{\partial y} & \dots \\ \frac{\partial H_1}{\partial y} & \frac{\partial H_1}{\partial x} & \frac{\partial H_2}{\partial y} & \frac{\partial H_2}{\partial x} & \dots \end{bmatrix} = \begin{bmatrix} H_1 \mathbf{A^T} & H_2 \mathbf{A^T} & \dots \end{bmatrix} \tag{A.56}$$

$$\mathbf{N} = \begin{bmatrix} N_1 & N_2 & \dots \\ N_1 & N_2 & \dots \\ 0 & 0 & \dots \end{bmatrix} \qquad \mathbf{M} = \begin{bmatrix} 2\mu & 0 & 0 \\ 0 & 2\mu & 0 \\ 0 & 0 & \mu \end{bmatrix} \qquad \mathbf{H} = \begin{bmatrix} H_1 & 0 \\ 0 & H_2 \\ \vdots & \vdots \end{bmatrix} \tag{A.57}$$

$$\mathbf{d} = \begin{pmatrix} u_1 \\ v_1 \\ \vdots \end{pmatrix} \qquad \mathbf{p} = \begin{pmatrix} p_1 \\ p_2 \\ \vdots \end{pmatrix} \qquad \mathbf{f} = \begin{pmatrix} 0 \\ -\rho g \end{pmatrix}. \tag{A.58}$$

The integrals are evaluated numerically using Gaussian integration points ip and weights l (Kwon and Bang, 2000):

$$-\sum_{ip} \mathbf{B^T N} \left|J\right| l \mathbf{p} + \sum_{ip} \mathbf{B^T M B} \left|J\right| l \mathbf{d} + \sum_{ip} \mathbf{H f} \left|J\right| l = 0. \tag{A.59}$$

Regarding the mass balance equation, some extra considerations are required. Since the pressure p does not occur in the equation of mass conservation, the mass balance merely puts constraints on the velocity solution that is sought. As was shown by Cuvelier et al. (1986), the force balance equation can be formulated as a minimization problem, and the compliance with the mass balance constraint $\boldsymbol{\nabla}_{\mathbf{x}} = 0$ is effected by imposing a penalty term. The penalty function method solves this problem by transforming the system of equations into an unconstrained minimization problem, which is solved iteratively. In the present case, a large penalty parameter τ is introduced, linking the pressure p to the divergence of the velocity:

$$\frac{1}{\tau} p = \boldsymbol{\nabla}_{\mathbf{x}} \cdot \mathbf{d}, \tag{A.60}$$

where τ is a large (penalty) parameter. The weak formulation of the weighted residual of the mass balance is then:

$$\frac{1}{\tau} \int_{\Omega_{el}} w\mathbf{n} \, d\Omega \, \mathbf{p} = -\int_{\Omega_{el}} w\mathbf{eB} \, d\Omega \, \mathbf{d}, \tag{A.61}$$

where

$$\mathbf{n} = \begin{pmatrix} N_1 \\ N_2 \\ \vdots \end{pmatrix}^T \qquad \mathbf{p} = \begin{pmatrix} p_1 \\ p_2 \\ \vdots \end{pmatrix} \qquad \mathbf{e} = \begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix}^T, \tag{A.62}$$

and B as before. Replacing the weighting function w by the pressure shape functions $\mathbf{n}$ according to Galerkin gives

$$\frac{1}{\tau} \int_\Omega \mathbf{G}\, d\Omega\, \mathbf{p} = -\int_\Omega \mathbf{N^T B}\, d\Omega\, \mathbf{d}, \tag{A.63}$$

$$\text{where } \mathbf{G} = \mathbf{nn}. \tag{A.64}$$

The same numerical integration scheme is applied as for the force balance equation:

$$\frac{1}{\tau} \sum_{ip} \mathbf{G}\, |J|\, l\, \mathbf{p} = -\sum_{ip} \mathbf{N^T B}\, |J|\, l \mathbf{d}, \tag{A.65}$$

The full system of equations of mass and linear momentum balance equations can now be written in matrix form:

$$\begin{bmatrix} E_{vv} & E_{vp} \\ E_{pv} & E_{pp} \end{bmatrix} \begin{pmatrix} \mathbf{d} \\ \mathbf{p} \end{pmatrix} = \begin{pmatrix} \mathbf{F} \\ \mathbf{0} \end{pmatrix}, \tag{A.66}$$

where

$$\mathbf{E_{vv}} = \sum_{ip} \mathbf{B^T M B}\, |J|\, l \tag{A.67}$$

$$\mathbf{E_{vp}} = -\sum_{ip} \mathbf{B^T N}\, |J|\, l \tag{A.68}$$

$$\mathbf{F} = -\sum_{ip} \mathbf{H f}\, |J|\, l \tag{A.69}$$

$$\mathbf{E_{pv}} = -\sum_{ip} \mathbf{N^T B}\, |J|\, l = \mathbf{E_{vp}}^T \tag{A.70}$$

$$\mathbf{E_{pp}} = \frac{1}{\tau} \sum_{ip} \mathbf{G}\, |J|\, l. \tag{A.71}$$

In this system, the pressure can be eliminated as follows:

$$\mathbf{E_{pv}d} + \mathbf{E_{pp}p} = 0 \tag{A.72}$$

$$\mathbf{p} = -\mathbf{E_{pp}^{-1}E_{pv}d} \tag{A.73}$$

$$\left(\mathbf{E_{vv}} - \mathbf{E_{vp}E_{pp}^{-1}E_{pv}}\right)\mathbf{d} = \mathbf{F}, \tag{A.74}$$

which is equivalent to eq. (8.1.22) in Cuvelier et al. (1986).

A.2.5 Powell/Hestenes iterations

Eq. (A.74) must be iterated in order to guarantee an accurate fulfilment of the incompressibility condition. Following the Powell & Hestenes method (Cuvelier et al., 1986), this is done in the following way:

$$\left(\mathbf{E_{vv}} - \mathbf{E_{vp}E_{pp}^{-1}E_{pv}}\right)\mathbf{d_{new}} = \mathbf{F} - \mathbf{E_{pv}p_{old}} \tag{A.75}$$

$$\mathbf{p_{new}} = \mathbf{p_{old}} + \tau\mathbf{E_{vp}d_{new}}, \tag{A.76}$$

where subscripts $_{new}$ and $_{old}$ indicate values from the previous and actual iteration step. This procedure allows the incompressibility condition to be approximated up to machine precision.

A.3 Dimensional analysis: mechanical model of power-law fluid

Dimensional analysis is a powerful tool to reduce the number of independent parameters of a given physical problem based on the principle of homogeneity of physical equations. Barenblatt (1996) gives a thorough introduction to the topic. The scaling which results from bringing all equations into nondimensional form can furthermore enhance the accuracy of the numerical calculations of Stokes Flow in finite element codes (Pelletier et al., 1989). For that reason, all calculations in the FLASH code are done in a nondimensional form. In the following, the nondimensionalization for the two-dimensional Stokes flow equations for two domains with different physical properties is described, pointing out the difficulties that arise when dealing with power-law viscous material.

A velocity-pressure formulation of the Stokes equations for two domains of a nonlinear fluid contains parameters of the following dimensions (Eqs. (A.28), (A.30), and (A.31)):

$$\begin{aligned}
[x] &= L \\
[u] = [v] &= L \cdot T^{-1} \\
[c_1] &= M \cdot L^{-1} \cdot T^{\left(\frac{1}{n_1}-2\right)} \\
[c_2] &= M \cdot L^{-1} \cdot T^{\left(\frac{1}{n_2}-2\right)} \\
[n_1] = [n_2] &= [] \\
[p] &= M \cdot T^{-2} L^{-1} \\
[\rho g] &= M \cdot L^{-2} \cdot T^{-2},
\end{aligned} \tag{A.77}$$

where L is the dimension of length, M of mass, and T of time. Note that the parameter c_i is not a quantity that is directly measurable in the laboratory, because its dimension is not a polynomial of integral multiples of the basic dimensions L, M and T. Instead, the unit of this parameter varies depending on the stress exponent n (see Turcotte and Schubert, 2002, pg. 321 for numerical examples).

Assuming the first three quantities x, u and c_1 to be independent, the quantities can be nondimensionalised in the following way (with $\dot{\epsilon_c} = u_c/x_c$, and $n_c = n_1$):

$$\begin{aligned}
x^* &= x/x_c \\
u^* &= u/u_c \\
c_1^* &= c_1/c_c \\
p^* &= p/\left(c_c \dot{\epsilon}_c^{\left(\frac{1}{n_1}\right)}\right) \\
c_2^* &= c_2/\left(c_c \dot{\epsilon}_c^{\left(\frac{1}{n_1}-\frac{1}{n_2}\right)}\right) \\
(\rho g)^* &= (\rho g)\,/\left(c_c x_c^{-1} \dot{\epsilon}_c^{\left(\frac{1}{n_1}\right)}\right)
\end{aligned} \tag{A.78}$$

where the numerical values of x_c, u_c and c_c can be arbitrarily chosen. As can be seen in Eqs. (A.78), the complications vanish if $n_1 = n_2$.

A.4 Derivation of the finite element formulation of the constitutive laws for viscous anisotropic material

This section is a commented listing of a Maple 8.0 file that performs the derivation of the finite element formulation of the constitutive equations for viscous (Newtonian or power-law) transversely anisotropic material in two dimensions, which were implemented in the FLASH code. In contrast to the isotropic case (Eq. (A.23)), the viscosity tensor Λ_{mnop} contains two different non-zero entries η and μ for the normal and shear viscosity respectively if the material is transversely anisotropic (incompressibility is assumed). These parameters describe the normal and shear viscosity parallel to a plane of anisotropy, which is itself described by the vector normal to it, the so-called 'director' $\vec{n}$ (following the nomenclature of Mühlhaus et al., 2002). The orientation of the plain of anisotropy at each material point can be described by the angle θ between the director $\vec{n}$ and the y-axis of the global coordinate system.

The parameters η and μ are given in the coordinate system $(\vec{s}, \vec{n})$, that is locally parallel/perpendicular to the plane of anisotropy. However, stresses and strainrates in the finite element code are calculated in global $(\vec{x}, \vec{y})$ coordinates. The tensor Λ_{mnop} must therefore be rotated into the global coordinate system, which will be done componentwise in the following (lines containing Maple commands start with an >).

Start with cleaning the workspace and loading the required packages:

```
> restart;
> with(LinearAlgebra):
> with(linalg):
```

The coordinates of the strain rate tensor $\dot{\epsilon}_{ij}$ are given in global coordinates $(\vec{x}, \vec{y})$. Because the material properties are described in local coordinates $(\vec{n}, \vec{s})$, the strain rate tensor must be rotated by an angle θ in order to calculate the local stresses. The rotation matrix R is defined as:

```
> R:=Matrix([[cos(theta),sin(theta)],[-sin(theta),cos(theta)]]):
```

However, R can also be expressed in terms of the components of the director $\vec{n}$ of the anisotropy: $n1 = -\sin\theta$, $n2 = \cos\theta$

```
> R:=Matrix([[n2,-n1],[n1,n2]]):
```

The global strain rate vector and matrix are initialized by:

```
> strr_v_xy:=Vector([epsilon11,epsilon22,epsilon12]):
> strr_xy:=Matrix([[strr_v[1],strr_v[3]],[strr_v[3],strr_v[2]]]):
```

The material matrix M is defined locally, i.e. with respect to the coordinates $(\vec{n}, \vec{s})$:

```
> M_ns:=Matrix([[2*eta,0,0],[0,2*eta,0],[0,0,2*mu]]):
```

where `eta` is the normal viscosity parallel to $\vec{n}$ or $\vec{s}$, and `mu` the shear viscosity.

Two auxiliary arrays are defined:

```
> R_transpose:=Matrix([[R[1,1],R[2,1]],[R[1,2],R[2,2]]]):
> epsi:=Vector([epsilon11,epsilon22,epsilon12]):
```

The strain rates in $(\vec{x}, \vec{y})$-coordinates can now be rotated to the $(\vec{n}, \vec{s})$ system:

```
> strr_ns:= R_transpose.strr_xy.R:
```

... and be reshaped to a vector:

```
> strr_v_ns:=Vector([strr_ns[1,1],strr_ns[2,2],strr_ns[1,2]]):
```

In the local $(\vec{n}, \vec{s})$ coordinate system, the stresses are calculated as:

```
> stress_v_ns:= Multiply(M,strr_v_ns):
```

... and can be reshaped to a matrix:

```
> stress_ns :=
> Matrix([[stress_v_ns[1],stress_v_ns[3]],[stress_v_ns[3],stress_v_ns[2]
> ]]):
```

The stress matrix is then rotated back into the $(\vec{x}, \vec{y})$ coordinate system:

```
> stress_xy:=simplify(R.stress_ns.R_transpose):
```

Reshape the stress matrix to a vector:

```
> stress_v_xy:=
> Vector([stress_xy[1,1],stress_xy[2,2],stress_xy[1,2]]):
```

Now the 'isotropic' part of the stress can be calculated and subtracted from the total stress vector:

```
> M_iso:= Matrix([[2*eta,0,0],[0,2*eta,0],[0,0,2*eta]]):
> stress_iso_v_xy:=M_iso.strr_v_xy:
> stress_aniso_v_xy := stress_v_xy-stress_iso_v_xy:
```

The coefficients of the strain rate components $\dot{\epsilon}_{ij}$ in the anisotropic part `stress_aniso_v_xy` of the constitutive equation can now be extracted:

```
> M_aniso:=Matrix([[0,0,0],[0,0,0],[0,0,0]]):
> for i from  1 to 3 do;
> for j from 1 to 3 do;
> M_aniso[i,j]:= factor(coeff(stress_aniso_v_xy[i],epsi[j])):
> end do;
> end do;
> L:
```

In the finite element formulation used in the FLASH code, the shear strain rate is formulated as:

$\dot{\epsilon}_{xy} := (\frac{\partial}{\partial y} u) + (\frac{\partial}{\partial x} v)$

...whereas we assumed throughout this analysis that

$\dot{\epsilon}_{xy} := 0.5 \left((\frac{\partial}{\partial y} u) + (\frac{\partial}{\partial x} v) \right).$

Therefore the third column of the matrices M_{iso} and M_{aniso} must be multiplied by 0.5:

```
> M_aniso[1..3,3] := L[1..3,3]*0.5:
> M_iso[1..3,3] := L[1..3,3]*0.5:
```

The finite element formulation of the constitutive equation for transversely anisotropic viscous material then reads:

$$\sigma_{ij} = \left(M_{iso} - (2\eta - 2\mu) \cdot M_{aniso} \right) \dot{\epsilon}_{ij}, \tag{A.79}$$

$$\text{where } M_{iso} = \begin{bmatrix} 2\eta & 0 & 0 \\ 0 & 2\eta & 0 \\ 0 & 0 & \eta \end{bmatrix}, \quad M_{aniso} = \begin{bmatrix} -\delta_0 & \delta_0 & -\delta_1 \\ \delta_0 & -\delta_0 & \delta_1 \\ -\delta_1 & \delta_1 & -\frac{1}{2} + \delta_0 \end{bmatrix}, \tag{A.80}$$

$\delta_0 = 2n_1^2 n_2^2$ and $\delta_1 = (n_1 n_2^3 - n_1^3 n_2)$. This formulation is equivalent to the one presented by Mühlhaus(2002a); the notations have been aligned for reasons of easy reference. In the case of three dimensions and/or more complicated material properties (e.g. orthotropic or monotropic behaviour), the demonstrated component-wise derivation soon becomes too circumstantial, and the use of the full tensor notation is then advised.

A.5 Code benchmarks

Even when taking utmost care in writing a numerical code, syntax errors or typing mistakes occur all too easily. It is therefore of key importance to test the code against analytical solutions in order to validate the results and to prove that the code is capable of reproducing standard modeling setups. In the following, three test setups are discussed against which the FLASH code was benchmarked. Testing the code against these benchmark examples provides a trustworthy basis for the interpretation of numerical results obtained in the frame of this thesis.

A.5.1 Test I: Folding of a Newtonian layer embedded in Newtonian matrix material

The setup of a linear viscous layer embedded in linear viscous material is well suited for the testing of numerical codes, since an analytical solution exist (Fletcher, 1974, 1977) which predicts the growth rate of an instability arising from an initial perturbation of the matrix-layer interface. Fletcher (1977) derived an analytical solution for this setup by linearising the complete thick-plate formulation of the problem. This analytical solution is suitable to test the precision and convergence of the numerical code. According to Fletcher (1977), the dimensionless growth rate α_d of a perturbed matrix-layer interface is given by

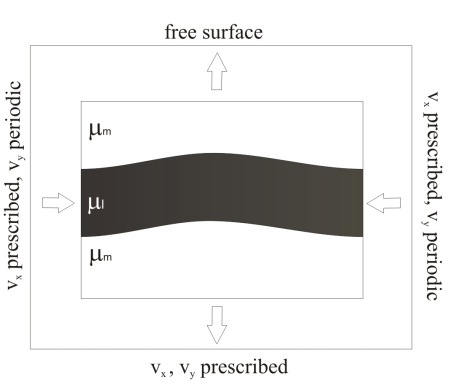
$$\alpha(k) = -2\,(1-R)\,\frac{1}{(1-R^2) - \frac{\left((1+R)^2 e^k - (1-R)^2 e^{-k}\right)}{2k}}, \tag{A.81}$$

where $\alpha(k)$ denote the growth rate, $R = \frac{\mu_m}{\mu_l}$ the inverse viscosity contrast between the layer and the matrix, $k = \frac{2\pi}{\lambda}h$ the dimensionless wave number, h is the thickness and λ the wavelength of the initial perturbation of the competent layer. The dominant wavelength, which grows fastest, maximizes Eq. (A.81).

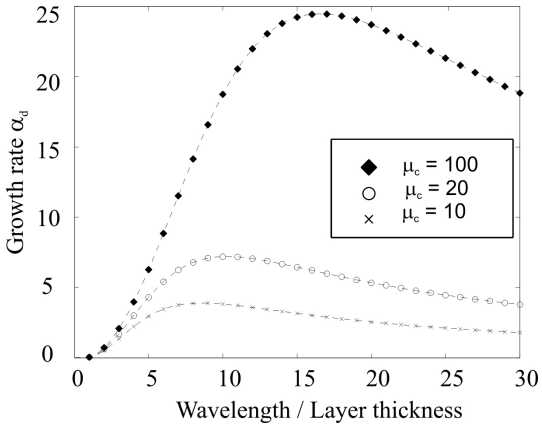
Fig. A.1(a) shows the setup of the test and the applied boundary conditions. Fig. A.1(b) displays the resulting growth rates α for viscosity contrasts of $10, 20$ and 100 between layer and matrix. The numerical growth rates match the analytical growth rates and converge toward the analytical solution for decreasing time step dt, decreasing amplitude of the initial perturbation, and increasing numerical resolution.

A.5.2 Test II: Folding of a power law viscous layer embedded in power law viscous matrix

The analytical solution for a layer of nonlinear viscous material embedded in nonlinear viscous matrix was also derived by Fletcher (1974). He performed a linear stability analysis of the thick plate formulation and developed a solution that describes the growth



(a) Setup of the folding benchmark test: a competent layer of viscosity μ_l is embedded in a matrix of lower viscosity μ_m. The layer–matrix interface is perturbed by a cosine function. A pure shear background flow field is applied.



(b) Numerical (symbols) vs. analytical (dashed lines) results for viscosity contrast of $\mu_c = 10, 20$ and 100. The numerical growth rates converge towards the analytical solution for decreasing initial perturbation amplitude, time step, and element size.

Figure A.1. *Folding benchmark test*

rate α of the layer–matrix interface:

$$\alpha(k) = 2n_L(1-R)\frac{1}{-(1-Q^2)+\sqrt{n_L-1}\frac{\left[(1-Q^2)e^{a_L k}-(1-Q^2)e^{-a_L k}\right]}{2\sin(\beta_L k)})}, \tag{A.82}$$

where again $R = \mu_M/\mu_L$ is the inverse viscosity contrast, $k = 2\pi/\lambda$ is the wave number, λ is the wavelength, and $Q = (n_L/n_M)^{1/2}R$. This setup is well suited to test a code that incorporates power law rheology in order to check the correctness of the implementation. The same setup as shown in Fig. A.1(a) was used with the only difference that two additional material constants n_L and n_M appear, which represent the power law stress exponents of the layer and the matrix respectively (Eq. (A.28)). Fig. A.2 displays the

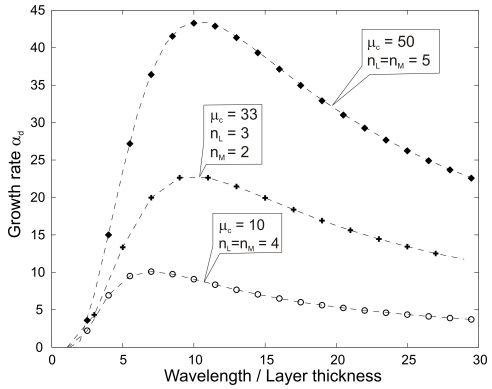


Figure A.2. *Nonlinear folding benchmark test*

numerical growth rates and the corresponding, analytically calculated growth rate for

different matrix/layer power law exponents and for different viscosity contrasts. The numerically calculated values correspond to the analytical values for the whole range of initial perturbation wavelengths. An important variation of the test presented above is the case in which unequal stress exponents n_L, n_M are used. In this case, an error in the nondimensionalisation of the code should become apparent, which is hidden in the case of equal exponents. The code nondimensionalisation applied to the FLASH code can be found in the appendix A.3.

The tests presented here focus primarily on the accuracy of the velocity fields for different rheologies. However, for geological problems it is of key importance that also stress and pressure fields are calculated correctly, as demonstrated in the following benchmark sample.

A.5.3 Test III: Stress distribution around a Newtonian elliptical inclusion embedded in Newtonian matrix material

Complex pressure and stress fields develop around an elliptical inclusion embedded in a Newtonian material when this material is subjected to a general shear flow (Schmid, 2002). The numerical results for stress and pressure fields obtained from a finite element approximation of this problem are very sensitive to the type of element that is chosen for the discretisation of the pressure p and the velocities u, v (Cuvelier et al., 1986). For example, the choice of linear approximations of both pressure and velocity leads to chessboard patterns in stress and pressure fields (Zienkiewicz and Taylor, 2000).

Schmid (2002) and Schmid and Podladchikov (2003) derived an analytical solution that describes pressure, stresses, and velocities of both matrix and inclusion for an arbitrary viscosity contrast. This analytical solution allows accurate benchmarking of the numerical results for this truly two-dimensional problem.

Fig. A.3 shows a comparison of numerically and analytically calculated values of the effective stress (second invariant of the stress tensor) in a matrix that surrounds a very thin weak elliptical inclusion (aspect ration $R = 150$), where the viscosity contrast between matrix and inclusion was $\mu_c = 1e5$. Seven-node enriched triangular Crouzeix-Raviart elements were used for the discretisation (Cuvelier et al., 1986).

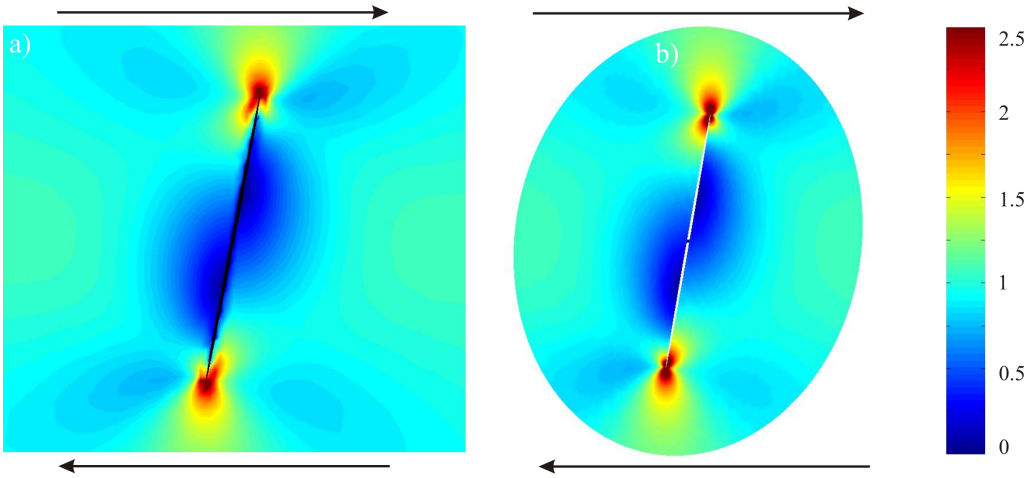


Figure A.3. *Effective stress field (= second invariant of the stress tensor as defined in Eq. (A.29)) in the matrix around a weak Newtonian inclusion embedded in Newtonian matrix material, subjected to a dextral simple shear flow. The effective stress values are normalized against the background effective stress. The numerical results are displayed in (a), the analytical results in (b). Length scales are missing because the problem was calculated in a nondimensional form. The differences in the effective stress field near the tips of the inclusion in (a) compared to (b) originate from the numerical resolution that cannot be made infinitesimally small. The ellipticity of both inclusion is* $R = 150$*, which is the approximate limit that can be successfully meshed by the MATLAB PDE Toolbox mesh generator that is used in the FLASH code.*